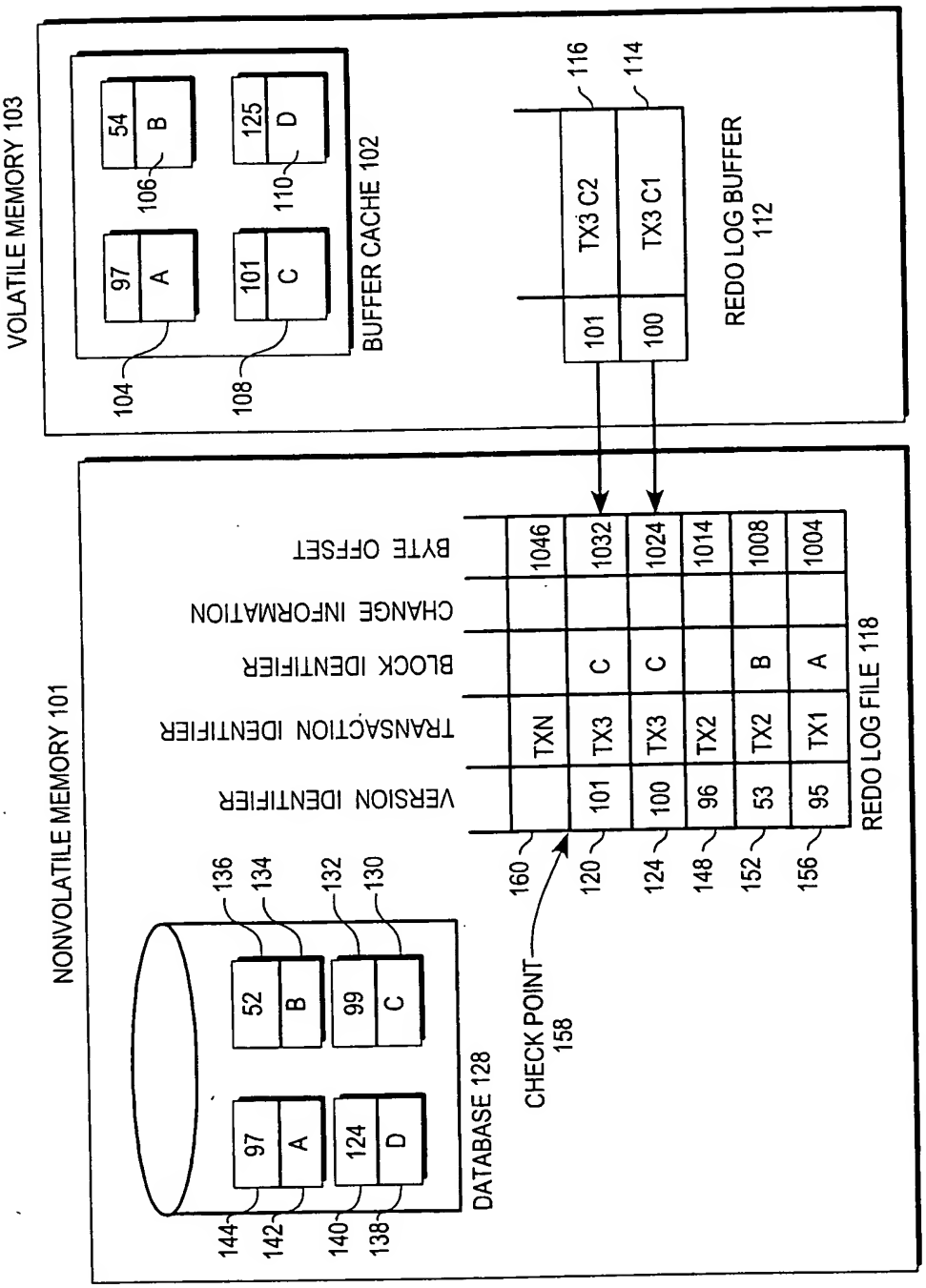

DATABASE SYSTEM 100
VOLATILE MEMORY 103
BUFFER CACHE 102
54 B
106
125 D
110
97 A
104
101 C
108
116
114
TX3 C2
TX3 C1
101
100
REDO LOG BUFFER 112
NONVOLATILE MEMORY 101
BYTE OFFSET
CHANGE INFORMATION
BLOCK IDENTIFIER
TRANSACTION IDENTIFIER
VERSION IDENTIFIER
1046
1032
1024
1014
1008
1004
TXN
TX3
TX3
TX2
TX2
TX1
C
C
B
A
101
100
96
53
95
160
120
124
148
152
156
REDO LOG FILE 118
CHECK POINT 158
136
134
132
130
52 B
99 C
97 A
124 D
144
142
140
138
DATABASE 128

Fig. 1

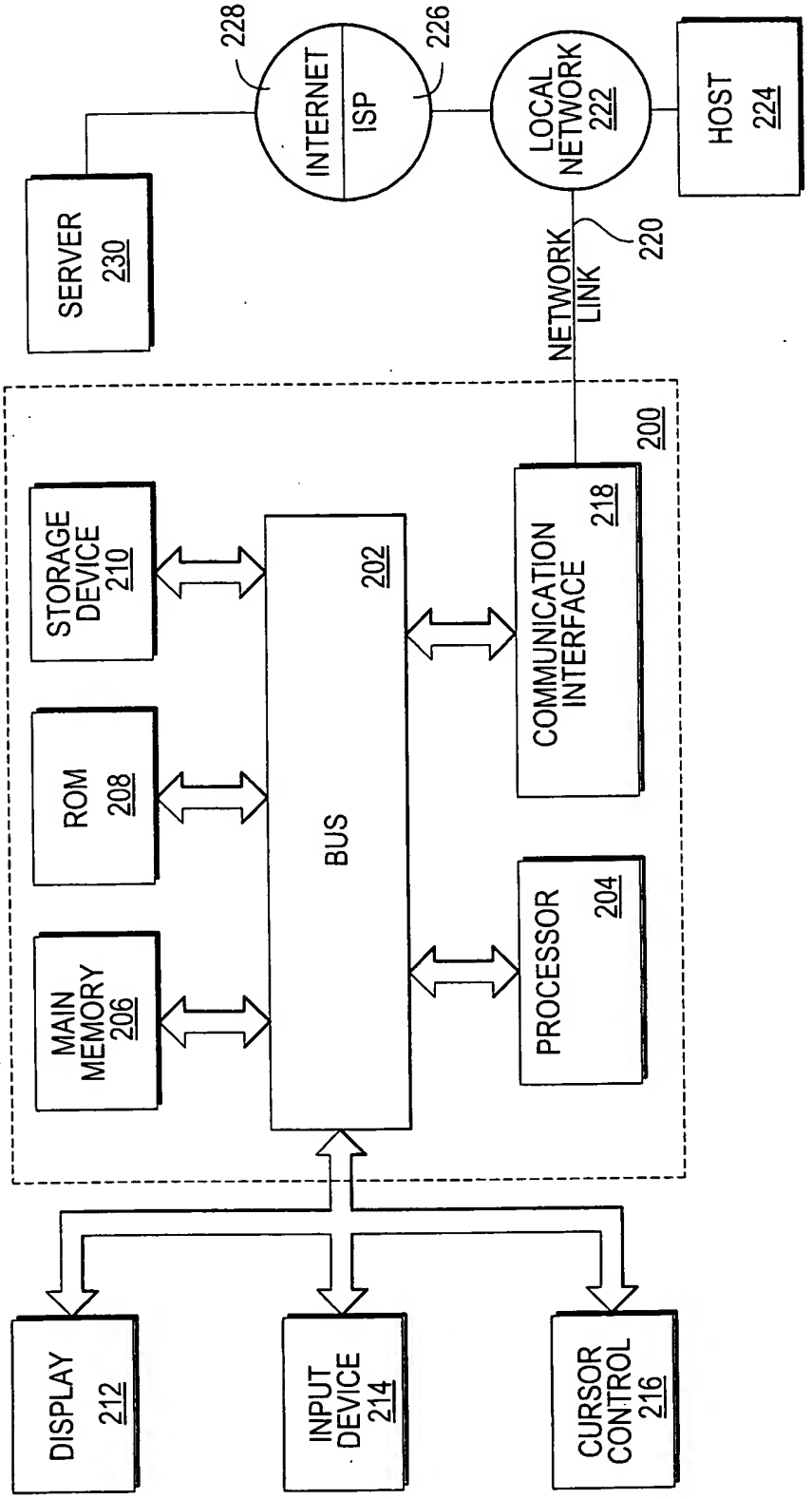

228
INTERNET
ISP
226
LOCAL NETWORK 222
HOST 224
SERVER 230
NETWORK LINK
220
200
STORAGE DEVICE 210
ROM 208
MAIN MEMORY 206
BUS 202
COMMUNICATION INTERFACE 218
PROCESSOR 204
DISPLAY 212
INPUT DEVICE 214
CURSOR CONTROL 216

Fig. 2

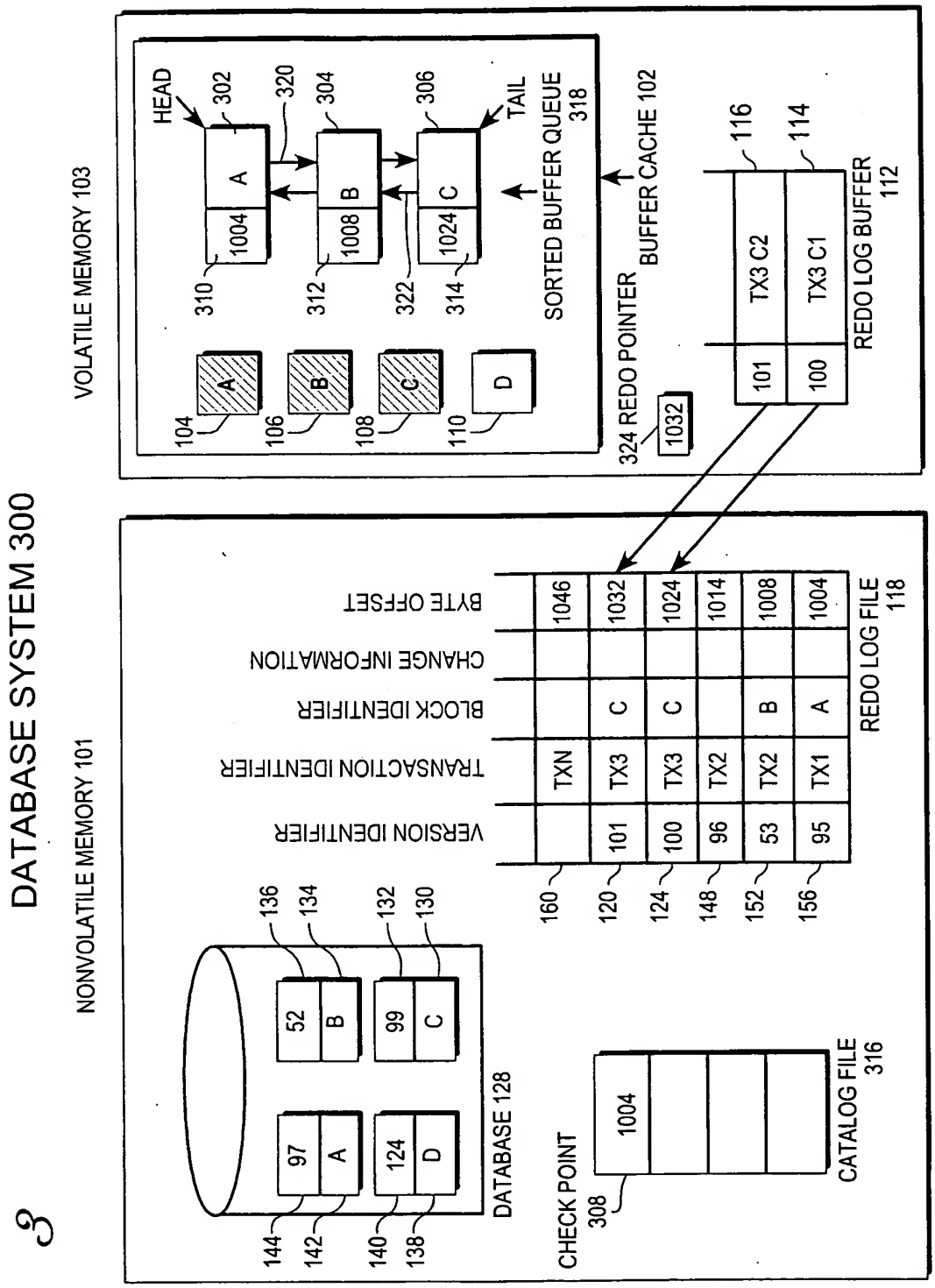

DATABASE SYSTEM 300
VOLATILE MEMORY 103
NONVOLATILE MEMORY 101
HEAD
TAIL
SORTED BUFFER QUEUE 318
BUFFER CACHE 102
REDO LOG BUFFER 112
TX3 C2
TX3 C1
324 REDO POINTER
REDO LOG FILE 118
BYTE OFFSET
CHANGE INFORMATION
BLOCK IDENTIFIER
TRANSACTION IDENTIFIER
VERSION IDENTIFIER
DATABASE 128
CHECK POINT 308
CATALOG FILE 316

Fig. 3

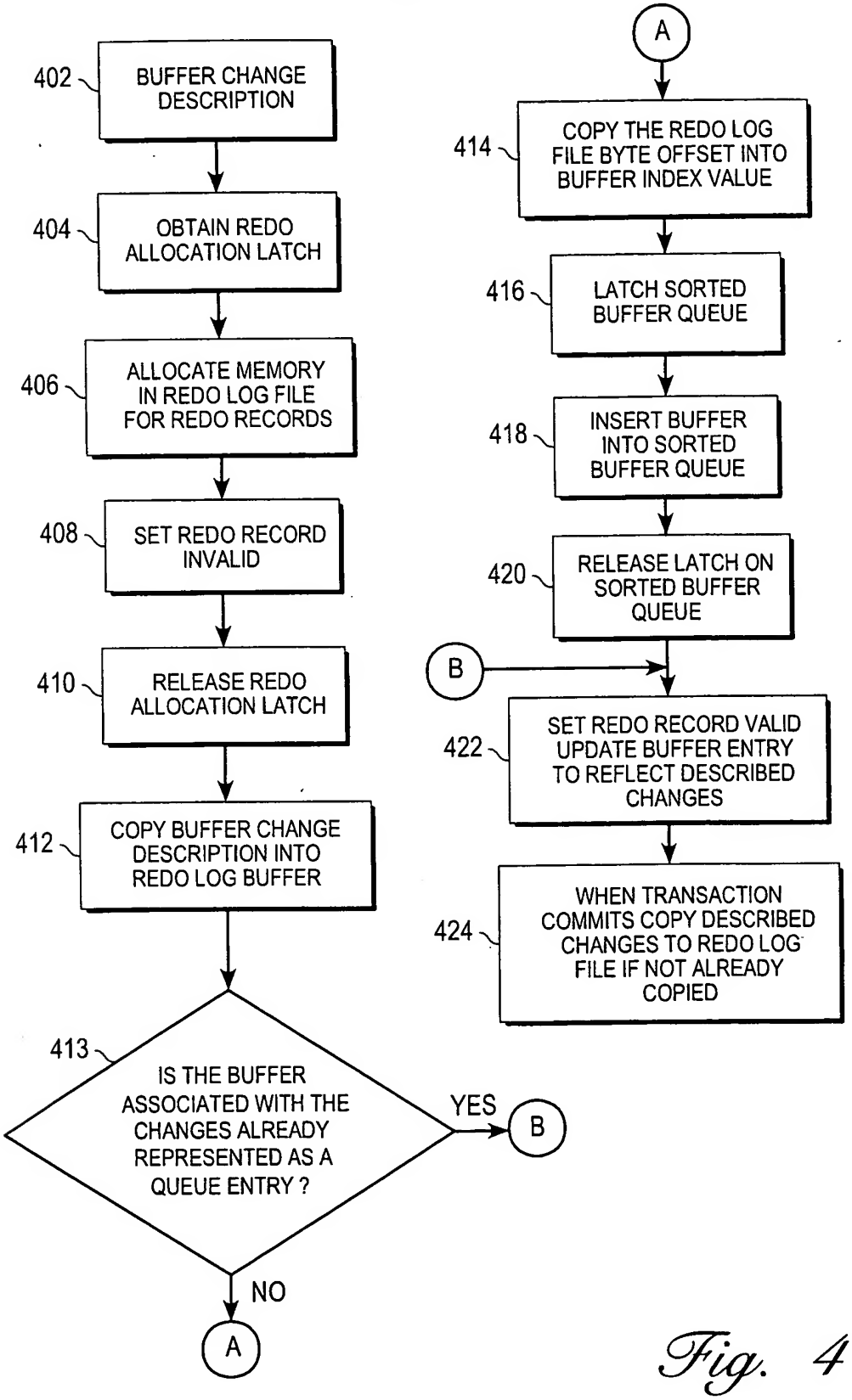

402
BUFFER CHANGE DESCRIPTION
404
OBTAIN REDO ALLOCATION LATCH
406
ALLOCATE MEMORY IN REDO LOG FILE FOR REDO RECORDS
408
SET REDO RECORD INVALID
410
RELEASE REDO ALLOCATION LATCH
412
COPY BUFFER CHANGE DESCRIPTION INTO REDO LOG BUFFER
413
IS THE BUFFER ASSOCIATED WITH THE CHANGES ALREADY REPRESENTED AS A QUEUE ENTRY ?
YES
B
NO
A
A
414
COPY THE REDO LOG FILE BYTE OFFSET INTO BUFFER INDEX VALUE
416
LATCH SORTED BUFFER QUEUE
418
INSERT BUFFER INTO SORTED BUFFER QUEUE
420
RELEASE LATCH ON SORTED BUFFER QUEUE
B
422
SET REDO RECORD VALID UPDATE BUFFER ENTRY TO REFLECT DESCRIBED CHANGES
424
WHEN TRANSACTION COMMITS COPY DESCRIBED CHANGES TO REDO LOG FILE IF NOT ALREADY COPIED

Fig. 4

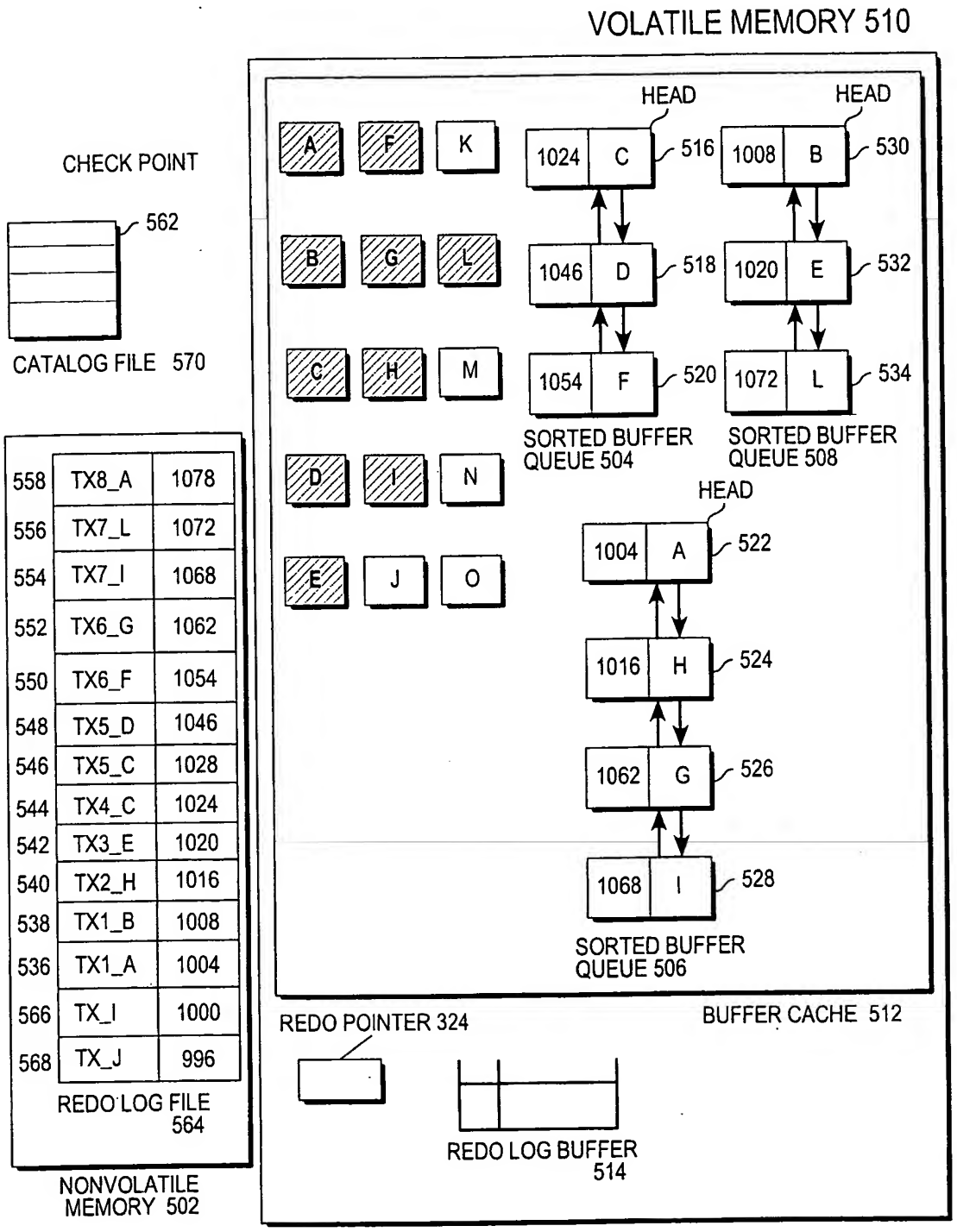

VOLATILE MEMORY 510
CHECK POINT
562
CATALOG FILE 570
558 TX8_A 1078
556 TX7_L 1072
554 TX7_I 1068
552 TX6_G 1062
550 TX6_F 1054
548 TX5_D 1046
546 TX5_C 1028
544 TX4_C 1024
542 TX3_E 1020
540 TX2_H 1016
538 TX1_B 1008
536 TX1_A 1004
566 TX_I 1000
568 TX_J 996
REDO LOG FILE 564
NONVOLATILE MEMORY 502
A F K
B G L
C H M
D I N
E J O
HEAD
1024 C 516
1046 D 518
1054 F 520
SORTED BUFFER QUEUE 504
HEAD
1008 B 530
1020 E 532
1072 L 534
SORTED BUFFER QUEUE 508
HEAD
1004 A 522
1016 H 524
1062 G 526
1068 I 528
SORTED BUFFER QUEUE 506
BUFFER CACHE 512
REDO POINTER 324
REDO LOG BUFFER 514

Fig. 5

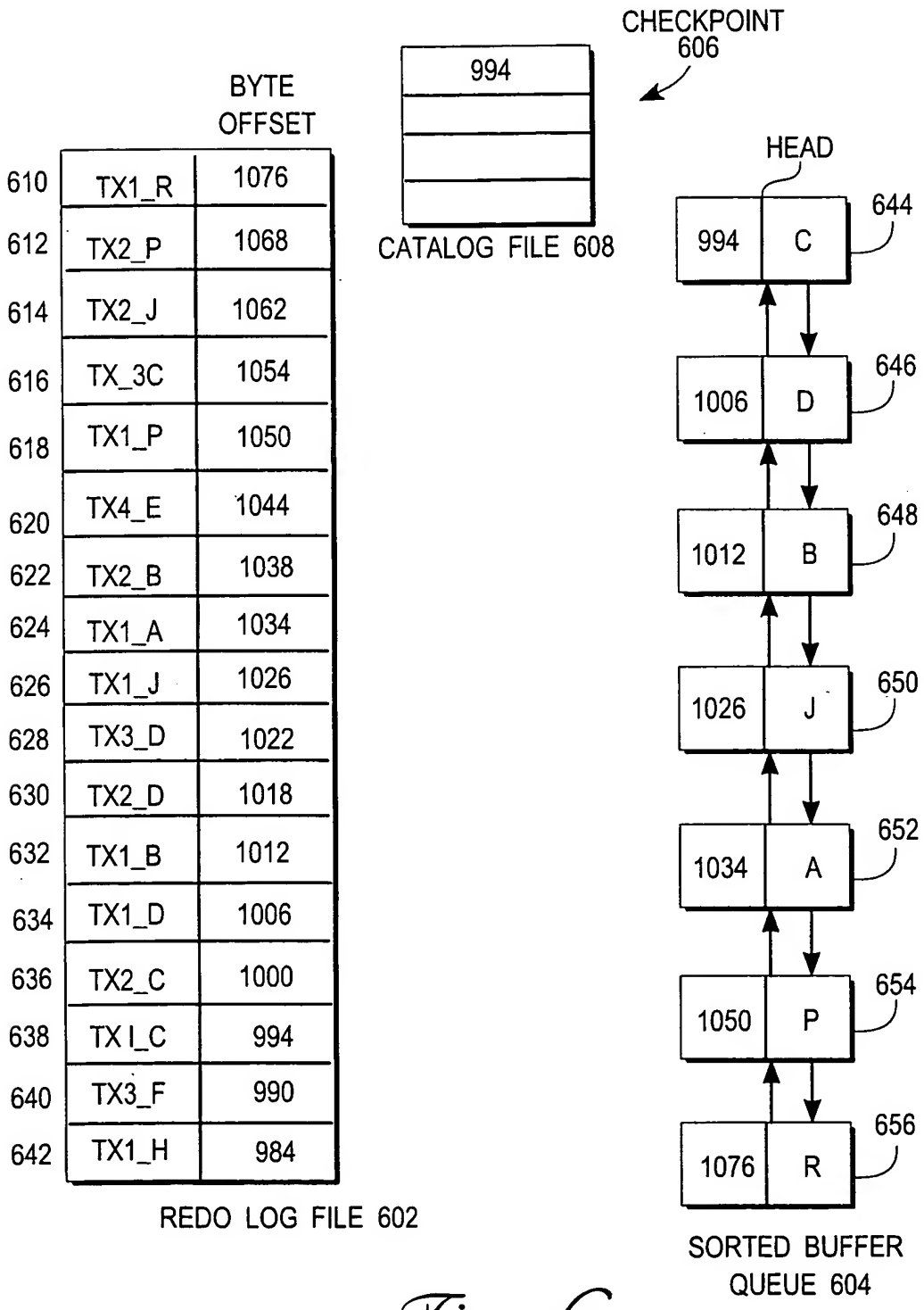

600
CHECKPOINT
606
994
CATALOG FILE 608
BYTE
OFFSET
610 TX1_R 1076
612 TX2_P 1068
614 TX2_J 1062
616 TX_3C 1054
618 TX1_P 1050
620 TX4_E 1044
622 TX2_B 1038
624 TX1_A 1034
626 TX1_J 1026
628 TX3_D 1022
630 TX2_D 1018
632 TX1_B 1012
634 TX1_D 1006
636 TX2_C 1000
638 TX I_C 994
640 TX3_F 990
642 TX1_H 984
REDO LOG FILE 602
HEAD
994 C 644
1006 D 646
1012 B 648
1026 J 650
1034 A 652
1050 P 654
1076 R 656
SORTED BUFFER
QUEUE 604

Fig. 6

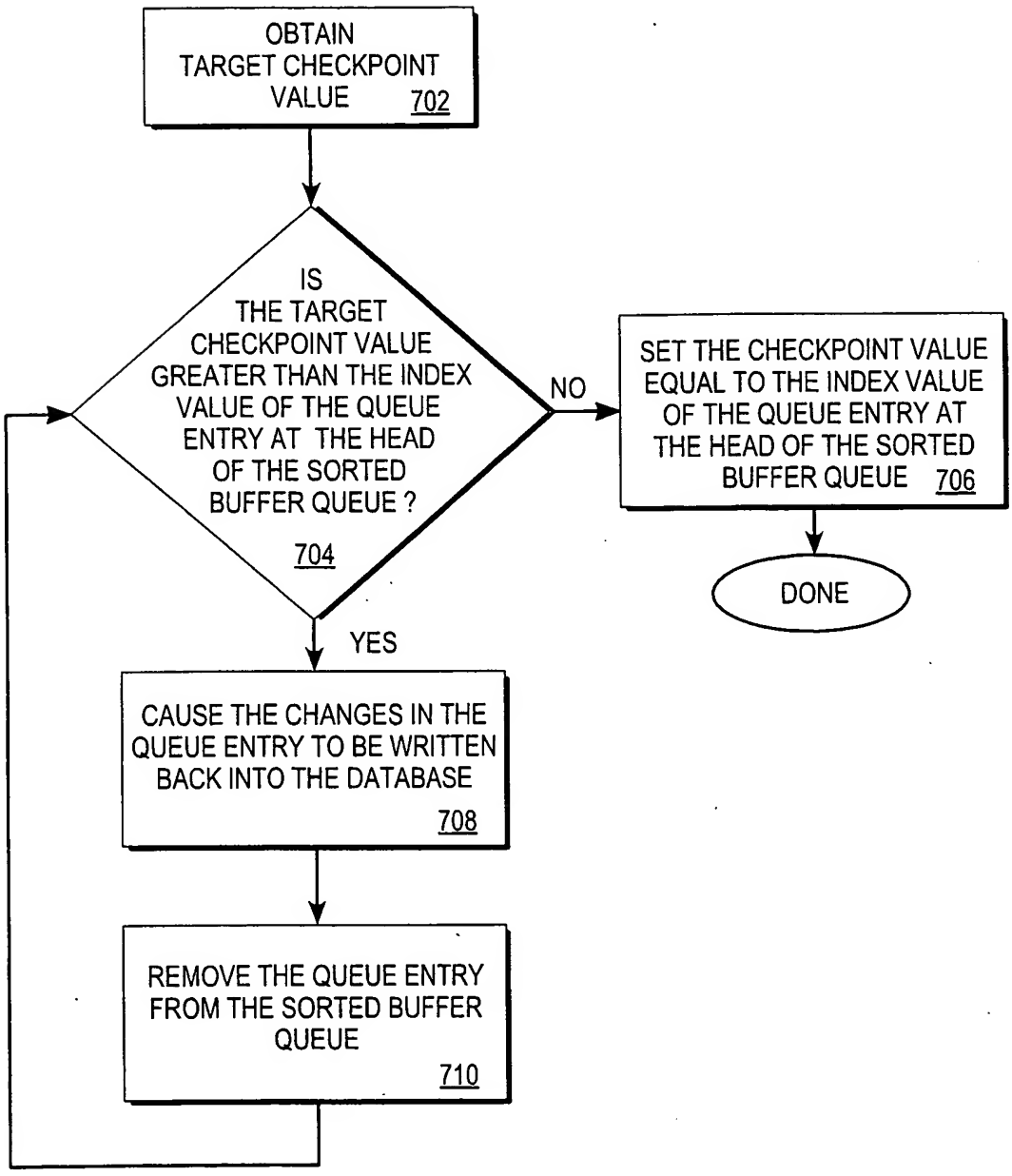

OBTAIN TARGET CHECKPOINT VALUE 702
IS THE TARGET CHECKPOINT VALUE GREATER THAN THE INDEX VALUE OF THE QUEUE ENTRY AT THE HEAD OF THE SORTED BUFFER QUEUE ? 704
NO
SET THE CHECKPOINT VALUE EQUAL TO THE INDEX VALUE OF THE QUEUE ENTRY AT THE HEAD OF THE SORTED BUFFER QUEUE 706
DONE
YES
CAUSE THE CHANGES IN THE QUEUE ENTRY TO BE WRITTEN BACK INTO THE DATABASE 708
REMOVE THE QUEUE ENTRY FROM THE SORTED BUFFER QUEUE 710

Fig. 7

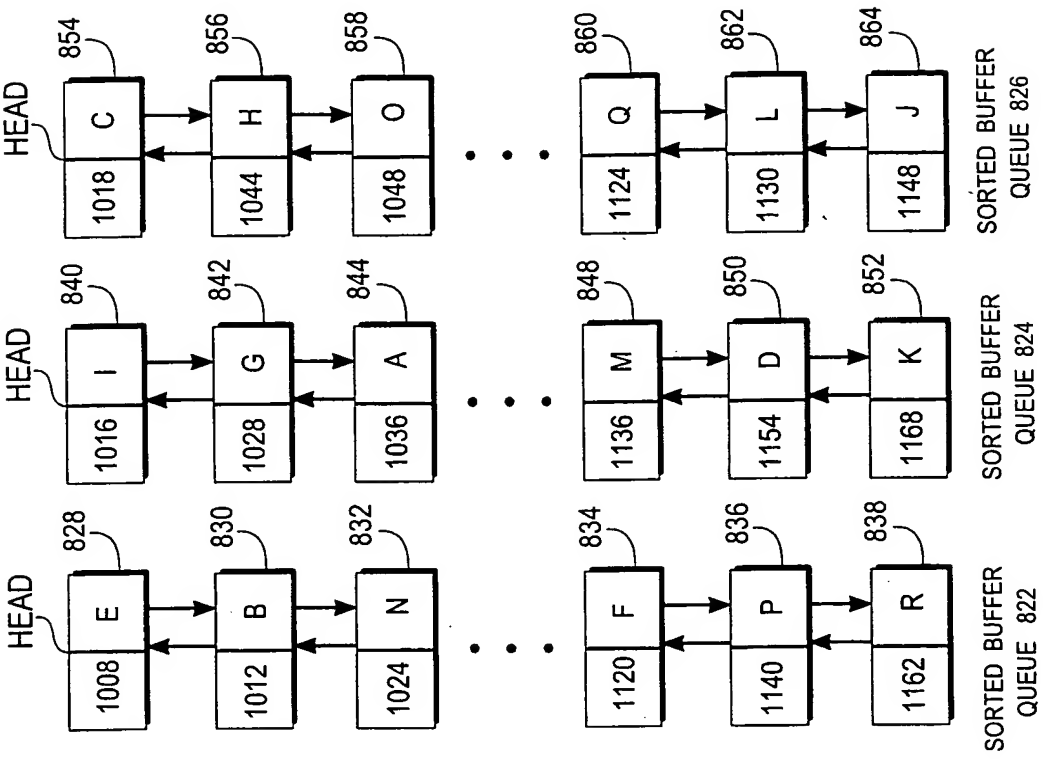
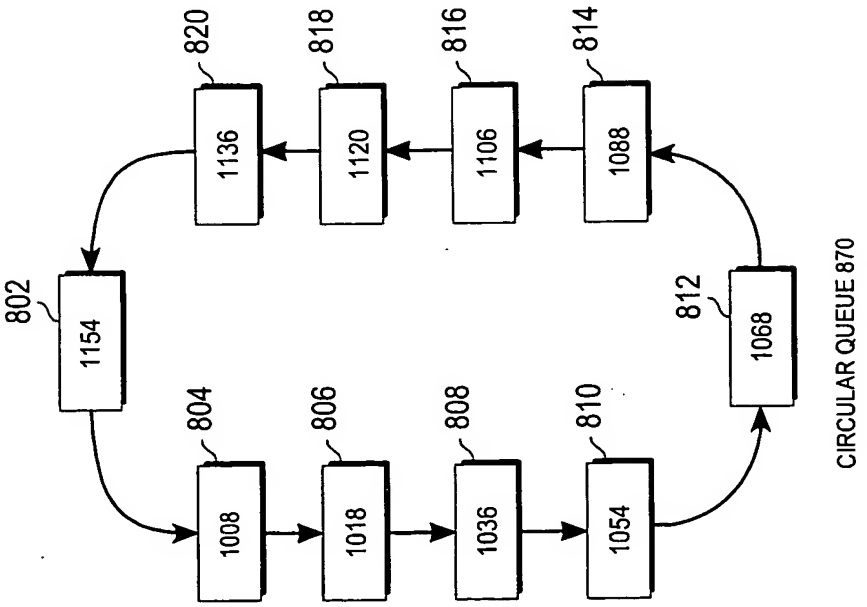

HEAD
854
C
1018
856
H
1044
858
O
1048
860
Q
1124
862
L
1130
864
J
1148
SORTED BUFFER QUEUE 826
HEAD
840
I
1016
842
G
1028
844
A
1036
848
M
1136
850
D
1154
852
K
1168
SORTED BUFFER QUEUE 824
HEAD
828
E
1008
830
B
1012
832
N
1024
834
F
1120
836
P
1140
838
R
1162
SORTED BUFFER QUEUE 822

Fig. 8


800
820
1136
818
1120
816
1106
814
1088
802
1154
812
1068
804
1008
806
1018
808
1036
810
1054
CIRCULAR QUEUE 870

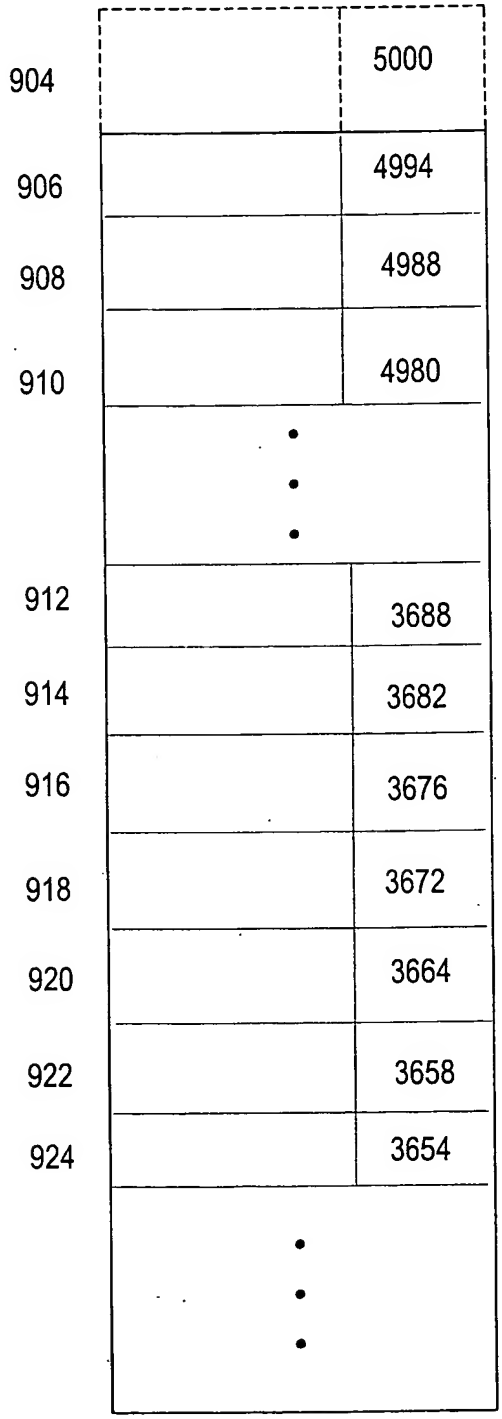

904
5000
906
4994
908
4988
910
4980
912
3688
914
3682
916
3676
918
3672
920
3664
922
3658
924
3654
REDO LOG FILE 902

Fig. 9